CC2511 Dongle

User Manual
Rev. 1.0

Table of contents

1 Introduction

Thank you for purchasing a SmartRF®04DK Development Kit.

Please note that Chipcon uses the term SmartRF®04DK as a collective term for all development kits on the SmartRF®04 platform. This development platform supports evaluation, development and prototyping using the SmartRF®04 products from Chipcon.

Make sure to subscribe to Chipcon's Developer's Newsletter to receive information about updates to documentation, new product releases and more. Sign up on the Chipcon web site.

2 About this manual

This manual covers the CC2511 Dongle, found in the CC2510–CC2511DK. It covers both the CC2511 Dongle hardware and software components of the development kit.
It covers the use of CC2511 application examples.
The CC2510EM and CC2510 application examples are covered in a separate document.
SmartRF® Studio has its own user manual.
The Chipcon Packet Sniffer has its own user manual.
The Chipcon Flash Programmer has its own user manual.
Further background information can be found in the device datasheets.

3 Definitions

SmartRF04EB	Evaluation Board. Main board with LCD, USB interface, LEDs, pot-meter etc. Platform for the Evaluation Modules (EM, see below).
SmartRF®04DK	A collective term used for all development kits for the SmartRF®04 platform, i.e. SmartRF04EB + EM.
CC2510EM	CC2510 Evaluation Module, a small plug-in module for SmartRF04EB, should be used as reference design for antenna and RF layout.
CC2511 USB Dongle	CC2511 Evaluation Module, a small USB dongle. SmartRF04EB, should be used as reference design for antenna and RF layout.
SoC	System on a Chip. A collective term used to refer to Chipcon ICs with on-chip MCU and RF transceiver. Used in this document to reference the CC2430 and the CC2510.
ICE	In Circuit Emulator.
CC2510 application examples	Application examples using only the CC2510.
CC2511 application examples	Application examples using the CC2511 USB Dongle together with a CC2510.

4 Connecting CC2511 Dongle to SmartRF04EB.

Please see the “CC2430DK_CC2510_CC1110_User_Manual” for more information on the SmartRF04EB and how to use the CC2510EM.
Make sure to install SmartRF® Studio **before** connecting the SmartRF04EB to a PC. SmartRF® Studio can be downloaded from the www.chipcon.com or www.ti.com/lpw

The CC2511 have a 2 wire debug interface that is used for chip programming and debugging. When connecting this interface to the SmartRF04EB USB MCU, the CC2511 can be programmed from the Chipcon Flash Programmer software and debugged from IAR embedded workbench..To connect the CC2511 Dongle to the SmartRF04EB follow these 6 steps:

1. Turn off the SmartRF04EB power by moving the power switch to the left position. See Figure 3.

2. Remove any CCxxxxEM modules attached to the SmarRF04EB.

3. Connect the SmartRF04EB to a PC with the supplied USB cable.

4. Connect the CC2511 Dongle to the SmartRF04EB with the supplied 10 pin cable.
   The cable should be mounted between the “SoC debug” connector on the CC2511 Dongle and the “P14 SoC debug / flash” connector on the SmartRF04EB. Make sure pin 1 is connected to pin 1. This cable connects the debug interface and GND between the two devices, however the CC2511 Dongle is not powered trough this cable.

5. Power the CC2511 Dongle.
   To power the CC2511 Dongle there are two options:

   - **Powered with a USB cable.**
     Using the supplied USB extension cable to connect the CC2511 Dongle to the PC.
     See Figure 1**.**

   - **Powered from the SmartRF04EB.**
     Connect pin 9 on the “Test pin” connector on CC2511 Dongle to pin 5 on the “I/O A” connector on SmartRF04EB to power the Dongle from the SmartRF04EB.
     See Figure 2.

   The CC2511 Dongle should only be powered by one of the two sources at a time.
   Do not connect the USB cable to the CC2511 Dongle while it is powered from the SmartRF04EB.
   The SmartRF04EB have a level converter on the debug connector, so the SmartRF04EB and the CC2511 Dongle may have a different power supply voltage.

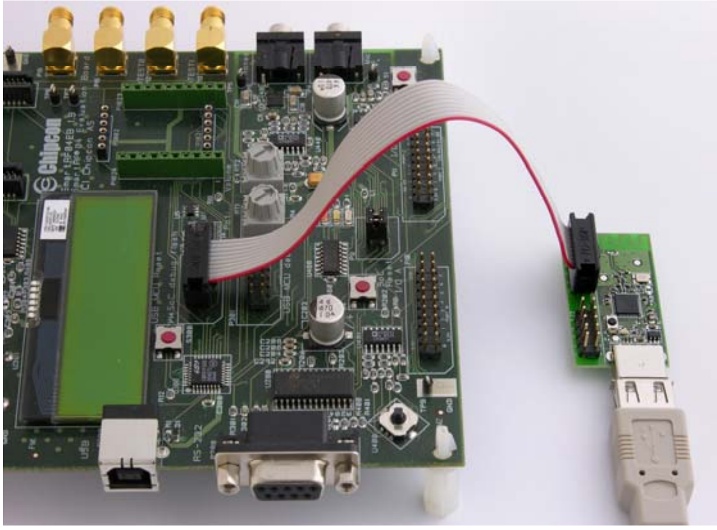


Figure 1 CC2511 Dongle powered from USB cable

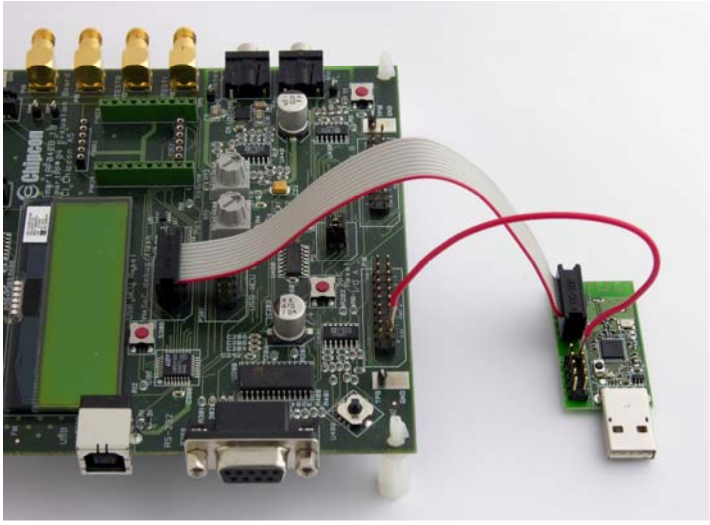


Figure 2 CC2511 Dongle powered from SmartRF04EB

6. Turn on the power on the SmartRF04EB. See Figure 3.

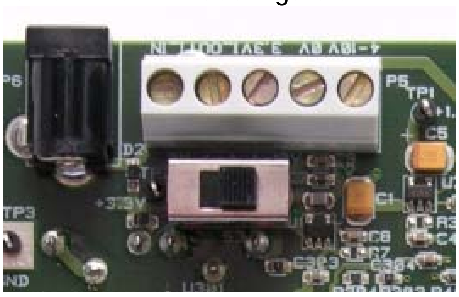


Figure 3: SmartRF04EB Power switch, power on.

The CC2511 can now be programmed with the Chipcon Flash Programmer software. The firmware on the CC2511 can also be debugged using the IAR embedded workbench debugger.
Please see the IAR user manual and the Chipcon Flash Programmer user manual for more details.

5 Using SmartRF04EB as an In-Circuit Emulator (ICE)

The SmartRF04EB can be used as ICE for the CC2511

To use the SmartRF04EB as ICE, the IAR Embedded Workbench software must be installed. The Embedded Workbench is a C-Compiler, Simulator, and ICE debugger. See the IAR Embedded Workbench documentation supplied with the kit for instructions how to set up the ICE debugger for use as an ICE.

When the SmartRF04EB is connected to a PC with the USB port, the debugger will connect to the SmartRF04EB. Several SmartRF04EB can be connected to USB ports simultaneously. A selection window will display the connected SmartRF04EB, and the user can select which device to load.

For custom PCB's with the CC2511, it is recommended to include a pin header or test points to allow in-circuit emulation or programming using SmartRF04EB or other 3rd party programming tools. Use the CC2511 Dongle as a reference. Please see the CC2510/CC2511section of the Chipcon web site for an updated list of 3rd party programming tools.

When using smartRF04EB as emulator for external target debugging any CCxxxxEM module must be removed

The pin-out used on the SmartRF04EB is explained in Table 1 and Figure 4. The connector includes 4 SPI control signals. These are currently not used, but they are included to give users flexibility.

Pin	Function
1	**Gnd**
2	**VDD**
3	**Debug Clock(DC)**
4	**Debug Data(DD)**
5	CSn(optional)
6	SCLK(optional)
7	**Reset_N**
8	MOSI(optional)
9	3.3V VDD, alt. NC
10	MISO

Table 1: P14 SOC debug connector pin-out

VDD note: The SmartRF04EB includes a voltage converter to support programming and debugging of external systems with different voltage than the SmartRF04EB. The debug connector includes two VDD connections on pin 2 and pin 9. The function is different for these connections.

Pin 2 VDD supplies voltage to the voltage converter. If the target application is self powered pin 2 should be connected to VDD to assure that the correct supply voltage is used for the voltage converter. This pin must always be connected to VDD.

Pin 9 VDD supplies 3.3V from the SmartRF04EB. If the target application is powered from the SmartRF04EB supply during programming and debugging this pin can be connected to VDD. If the target voltage differs from 3.3V, this pin should not be connected.
On the CC2511 Dongle this pin is not connected. Hence the CC2511 Dongle must be powered from a separate USB cable or a separate cable connecting pin 9 on “test pins” / P4 to VDD on the SmartRF04EB. See chapter 4 for details.

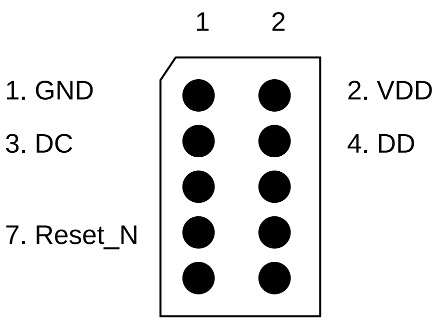


Figure 4: Recommended debug connector layout (Top view)

Figure 4 shows the required signal for a minimum connector layout.

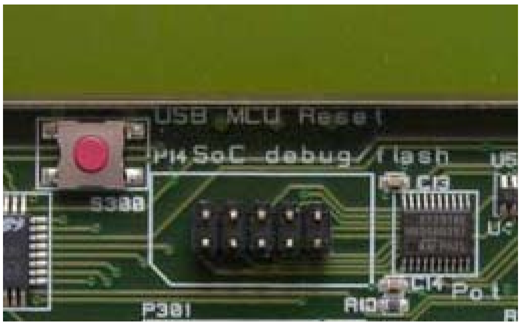


Figure 5: SmartRF04EB debug connector for debugging CC2511 Dongle / other external target systems

5.1 Debugging USB firmware

When writing and debugging USB firmware one should remember that when Windows (or another OS) detects a USB device it will assume that it works in accordance with the USB specification. During debugging this might not be the case. Breakpoints may stop the execution of code on the CC2511 Dongle and firmware under development may not be completed or error free. This may lead to the PC disconnecting the CC2511 Dongle and turning off or toggling its power supply. In some cases Windows (or the OS in use) may crash or misbehave. Hence breakpoints should be used with caution.

Obviously inserting and using breakpoints in the firmware during debugging will be extremely useful to pinpoint bugs and follow code execution. And they can and should be used. But they will also stop the code execution on the CC2511 and may cause the device to act in conflict with the USB specification. This can lead to the PC disconnecting the CC2511 Dongle, sending a USB reset or worst case the operating system may crash.

If the firmware contain bugs that make the PC disconnect the CC2511 Dongle it may be necessary to disconnect the USB cable, and power the CC2511 Dongle from the SmartRF04EB in order to reprogram it.

5.2 Known problems

When using the debug interface through the level-converter on the SmartRf04EB. (Shown in Figure 1 and Figure 2) it might become unstable. This problem is more likely to occur a low voltages or with crosstalk in the cable at high interface speed.

First make sure that the 10 pin flat cable used to connect the CC2511 USB Dongle to the SmartRF04EB is as short as possible.
Next it might be necessary to reduce the debug interface speed in order to achieve stable operation.
In the IAR workbench do the following:

- Highlight the project name by clicking on it in the "Files" window. (See Figure 6)
- Go to Project -> Options…
- Select the Category "Chipcon"
- Click on the "Target" tab.
- Check the "Reduce interface speed" option. (See Figure 7)

Highlight the project name

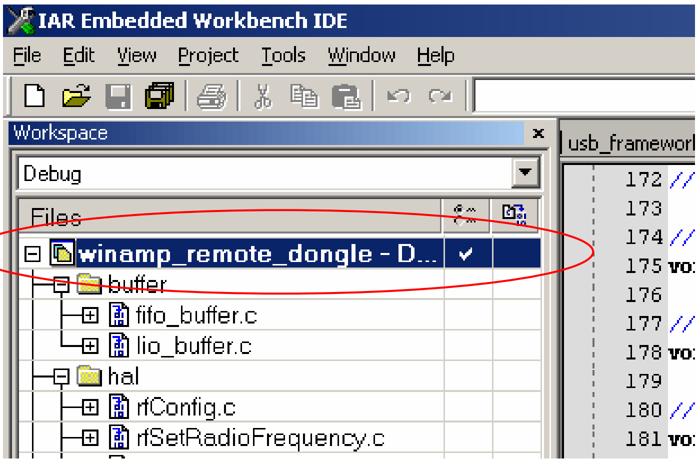
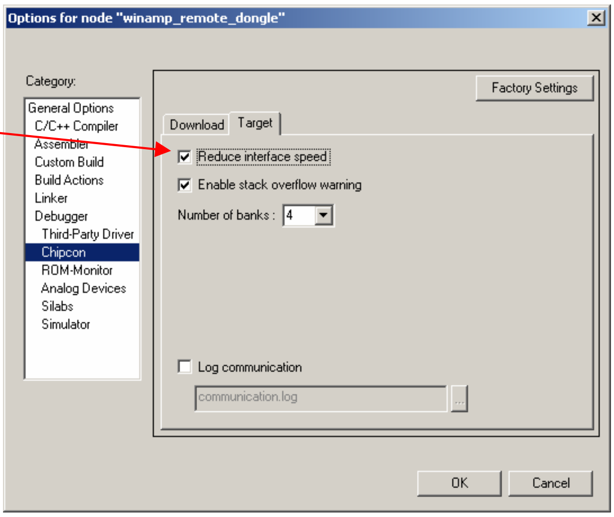


Figure 6: Highlight the project name

Check the "Reduce interface speed" option.



☑ Reduce interface speed

☑ Enable stack overflow warning

Number of banks : 4

□ Log communication

communication.log

Figure 7: Reduce interface speed

6 CC2511 application examples

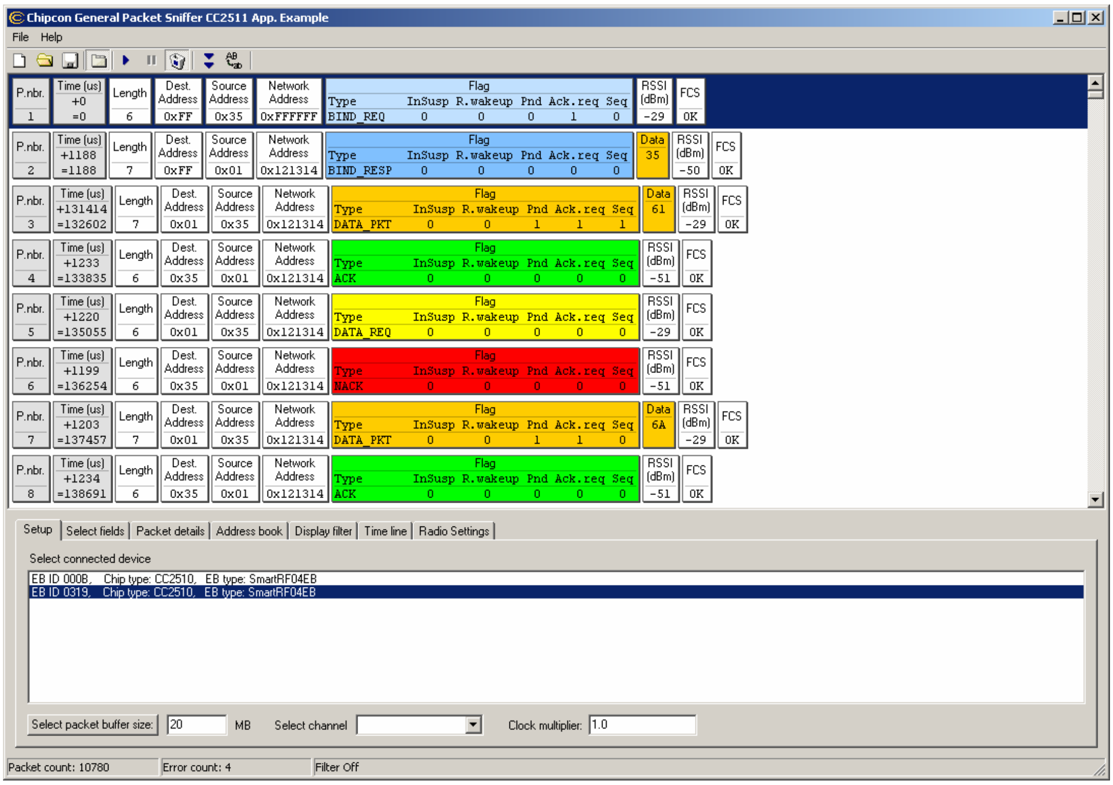
This section describes the application examples available for the CC2511. These application examples focus on how to use the USB on the CC2511. The examples is available both as source code and ready-to-upload .hex files from the Chipcon website.

There are three CC2511 application examples.

• Wireless mouse / practical joke.	(HID , Human Interface Devices)
• Wireless USB to Serial port converter.	(CDC, Communication Devices)
• Wireless WinAmp remote control.	(Vendor requests, custom Windows driver)

There is also a CC2510 Application Example available. This example provides valuable information about setup and usage of different modules found in both CC2510 and CC2511. Please see the “CC2430DK_CC2510_CC1110_User_Manual” for more information about this Application Example.

All the application examples use the same packet format and radio frequency. This is done to make it simple to observe the radio activity on the Chipcon General Packet Sniffer. After installing the packet sniffer choose the “CC2511 App. Example protocol” to see the radio traffic. Figure 8 show the packet sniffer observing radio traffic in the “Wireless USB to Serial port converter” application example.
See the Chipcon General Packet Sniffer user manual for more information.



P.nbr.	Time (us)	Length	Dest. Address	Source Address	Network Address	Type	InSusp	R.wakeup	Pnd	Ack.req	Seq	Data	RSSI (dBm)	FCS
1	+0 =0	6	0xFF	0x35	0xFFFFFF	BIND_REQ	0	0	0	1	0		-29	OK
2	+1188 =1188	7	0xFF	0x01	0x121314	BIND_RESP	0	0	0	0	0	35	-50	OK
3	+131414 =132602	7	0x01	0x35	0x121314	DATA_PKT	0	0	1	1	1	61	-29	OK
4	+1233 =133835	6	0x35	0x01	0x121314	ACK	0	0	0	0	0		-51	OK
5	+1220 =135055	6	0x01	0x35	0x121314	DATA_REQ	0	0	0	0	0		-29	OK
6	+1199 =136254	6	0x35	0x01	0x121314	NACK	0	0	0	0	0		-51	OK
7	+1203 =137457	7	0x01	0x35	0x121314	DATA_PKT	0	0	1	1	0	6A	-29	OK
8	+1234 =138691	6	0x35	0x01	0x121314	ACK	0	0	0	0	0		-51	OK

Select connected device

EB ID 000B, Chip type: CC2510, EB type: SmartRF04EB
EB ID 0319, Chip type: CC2510, EB type: SmartRF04EB

Select packet buffer size: 20 MB

Select channel

Clock multiplier: 1.0

Packet count: 10780 Error count: 4 Filter Off

Figure 8: Screenshot of the Chipcon General Packet Sniffer

6.1 Wireless mouse / practical joke

This application demonstrates the use of the CC2511Dongle as a USB composite HID (Human Interface Device), with both keyboard and mouse.
The application will work on most operating systems.

This application example uses the USB framework found in the CC2511 firmware library to setup the USB on the CC2511 Dongle and to handle all USB standard requests. In addition the Class requests required by the HID device class are implemented in a separate file.
The radio communication between CC2510EM and CC2511 Dongle is done using the RF part of the CC2511 firmware library.

See “Device Class Definition for Human Interface Devices (HID)” found on http://www.usb.org/developers/devclass_docs/HID1_11.pdf for more information about the HID class.

6.1.1 Software components

- **simple_hid_dongle.hex**
  The CC2511 USB Dongle firmware, is found in
  ”CC2511DK\embedded_sw\App_Ex\simple_hid\simple_hid_dongle\”.

- **simple_hid_eb.hex**
  The CC2510 EM firmware is found in
  ”CC2511DK\embedded_sw\App_Ex\simple_hid\simple_hid_eb\”.

6.1.2 Installing the application

1. Attach the CC2510EM to a SmartRF04EB and program it with the flash image simple_hid_eb.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See “CC2430DK_CC2510_CC1110_User_Manual” for more information).

2. Connect the CC2511 Dongle to the other SmartRF04EB and program it with the flash image simple_hid_dongle.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See “CC2430DK_CC2510_CC1110_User_Manual” for more information).

6.1.3 Running the application

Insert the CC2511 Dongle into a free USB port. The firmware will identify the CC2511 Dongle as a composite HID with a keyboard and mouse.
On most operating systems no extra driver is necessary; the OS will automatically load a standard driver for HID devices.

The pot-meter “RT1” on the SmartRF04EB is used to choose between the “simple mouse” and the “practical joke” mode. Turn the pot-meter fully counter clockwise to select the “simple mouse”, fully clockwise to select the “practical joke” mode.

In “simple mouse” mode the joystick on the SmartRF04EB can be used as a mouse. The button “S1” is the left mouse button, pushing the joystick is the right mouse button.
While in simple mouse mode the CC2510 will calculate the RSSI (Receive Signal Strength Indicator) on the packets received from the CC2511 Dongle and display it on the LCD screen. The RSSI value is updated each second.

In “practical joke” mode there are three jokes to choose from:

- Moving mouse – the mouse will constantly move slowly towards the right.
- Write to screen – Writes the text “anybody home” to the PC.
- Change window (ALT – TAB) – Changes the active window in Windows by sending the ALT – TAB keystrokes.

Move the joystick up and down to select joke, push “S1” button to start/stop.

6.2 Wireless USB to Serial port converter

This application example is a wireless USB to serial port (RS232) converter.
It creates a virtual serial port in Windows. Data sent to this virtual serial port will be sent to the CC2511 Dongle over USB, then to CC2510 EM over radio and finally out on the serial port on the CC2510 EM / SmartRF04EB.
Data sent to the serial port on the SmartRF04EB will be transferred to the virtual serial port in Windows.
Full duplex and flow control on the serial port (CTS and RTS lines) are supported. The radio link has CRC check and automatic retransmission for error free operation.

The application uses a CDC driver found in Windows, if used on other operating systems a suitable driver must be found/created. The Windows driver is rather slow and will limit the maximum throughput to approx 70 Kbit/s.

This application example uses the USB framework found in the CC2511 firmware library to setup the USB on the CC2511 Dongle and to handle all USB standard requests. In addition the class requests required by the CDC device class are implemented in a separate file.
The radio communication between CC2510EM and CC2511 Dongle is done using the RF part of the CC2511 firmware library.
The application example also uses the buffer module in the library to handle data buffering and the timer4 manager module for timing.

See "Class Definitions for Communication Devices" found on http://www.usb.org/developers/devclass_docs/usbcdc11.pdf for more information about the CDC class.

6.2.1 Software components

- **rs232_to_usb_dongle.hex**
  The CC2511 USB Dongle firmware, is found in
  "CC2511DK\embedded_sw\App_Ex\rs232_to_usb\rs232_to_usb_dongle\".

- **rs232_to_usb_eb.hex**
  The CC2510 EM firmware is found in
  "CC2511DK\embedded_sw\App_Ex\rs232_to_usb\rs232_to_usb_eb\".

- **CC2511ApExSerialPort.inf**
  The windows driver for the Wireless USB to Serial port, found in
  "CC2511DK\pc_sw\App_Ex\rs232_to_usb\driver\".

6.2.2 Installing the application

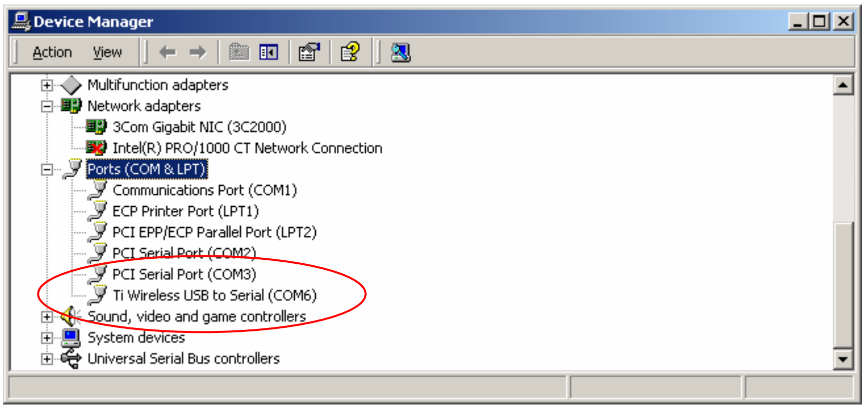
1. Attach the CC2510EM to a SmartRF04EB and program it with the flash image rs232_to_usb_eb.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See "CC2430DK_CC2510_CC1110_User_Manual" for more information).

2. Connect the CC2511 Dongle to the other SmartRF04EB and program it with the flash image rs232_to_usb_dongle.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See "CC2430DK_CC2510_CC1110_User_Manual" for more information).

3. Disconnect the programmed CC2511 USB Dongle from the SmartRF04EB and plug it into a free USB port on the PC.

4. When windows ask for a driver, point to the file "CC2511ApExSerialPort.inf".

When Windows have finished installing the device, a new serial port will appear in the device manager in Windows. See Figure 9.



Multifunction adapters
Network adapters
3Com Gigabit NIC (3C2000)
Intel(R) PRO/1000 CT Network Connection
Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
PCI EPP/ECP Parallel Port (LPT2)
PCI Serial Port (COM2)
PCI Serial Port (COM3)
Ti Wireless USB to Serial (COM6)
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 9: Virtual serial port in Windows Device manager

The Windows driver will only work if the “Ti Wireless USB to Serial” is assigned to COM1 to COM9. (In Figure 9 it is assigned to COM6)
If there are many COM ports installed on the PC, Windows may chose to assign it to a number higher than 9. If that happens it must be changed. Right click on the “Ti Wireless USB to Serial” in the device manager and choose “Properties” -> “Port settings” -> “Advanced” to change.

6.2.3 Running the application

Connect the CC2510EM to the SmartRF04EB and power it. Use the joystick and the S1 button to select the settings for the UART. The settings can only be selected once, if they are wrong, turn power off and on to reset the CC2510EM and select UART settings again.
These settings only apply to the UART on the CC2510/SmartRF04EB, the settings used on the Virtual COM port on the PC can be different.
Changing the settings on the CC2510EM/SmartRF04EB will not affect the settings used in the Virtual COM port and visa versa.

To bind the CC2510EM to the CC2511 Dongle, hold them close together and push S1 on the SmartRF04EB. The RF protocol found in the CC2511 firmware library uses kiss-binding which mean that the CC2511 Dongle use the RSSI (Received Signal Strength Indicator) to only accept bind requests from units that are in close proximity. If the two devices are too far apart (more than approx 1 meter) the CC2511 will not accept the bind request.

To test the link connect a RS232 cable between a PC and the SmartRF04EB, start HyperTerminal in Windows (Start -> Programs -> accessories -> Communications -> HyperTerminal)
Connect one HyperTerminal window to the Virtual COM port, and another HyperTerminal window to the COM port connected to the CC2510EM/SmartRF04EB.
Text written to one window will be sent to the other one.

6.3 Wireless WinAmp remote control

This application example is a wireless remote control for WinAmp media player. It is a two-way remote, which display the song information of the song played on the LCD screen on the SmartRF04EB.
The communication between the CC2511 Dongle and WinAmp is handled by a custom WinAmp plug-in. Hence this remote control will work with WinAmp only.

This application example uses the USB framework found in the CC2511 firmware library to setup the USB on the CC2511 Dongle and to handle all USB standard requests. In addition it demonstrates the use of Vendor requests in the USB framework. On the PC the CUDAL library is used to create a custom driver (see chapter 7.2 Windows USB driver framework). The radio communication between CC2510EM and CC2511 Dongle is done using the RF part of the CC2511 firmware library.
The application example also uses the buffer module in the library to handle data buffering and the timer4 manager module for timing.

6.3.1 Software components

- **winamp_remote_dongle.hex**.
  The CC2511 USB Dongle firmware, is found in
  "CC2511DK\embedded_sw\App_Ex\winamp_remote\winamp_remote_dongle\",

- **winamp_remote_eb.hex**
  The CC2510 EM firmware, is found in
  "CC2511DK\embedded_sw\App_Ex\winamp_remote\winamp_remote_eb\"

- **gen_chipcon_remote.dll**
  The WinAmp plug-in, is found in
  "CC2511DK\pc_sw\App_Ex\winamp_remote\WinAmp_pluginn"

- **CC2511ApExWinAmpRemote.inf, cudal.sys and cudal98.sys**
  The windows driver for the WinAmp remote control, found in
  "CC2511DK\pc_sw\App_Ex\winamp_remote\driver"

6.3.2 Installing the application

1. Attach the CC2510EM to a SmartRF04EB and program it with the flash image winamp_remote_eb.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See "CC2430DK_CC2510_CC1110_User_Manual" for more information).

2. Connect the CC2511 Dongle to the other SmartRF04EB and program it with the flash image winamp_remote_dongle.hex, using IAR workbench or the Chipcon FLASH programmer.
   (See "CC2430DK_CC2510_CC1110_User_Manual" for more information).

3. Download WinAmp from: www.winamp.com

4. Install WinAmp.

5. Install the WinAmp plug-in by copying the file "gen_chipcon_remote.dll" to the "C:\Program Files\Winamp\Plugins\" folder.

6. Disconnect the programmed CC2511 USB Dongle from the SmartRF04EB and plug it into a free USB port on the PC.

7. When windows ask for a driver, point to the file "CC2511ApExWinAmpRemote.inf".

8. If windows ask for the file “cudal.sys” or “cudal98.sys”, point to “cudal.sys” or “cudal98.sys”.

9. Turn on the CC2510EM / SmartRF04EB.

10. The USB dongle may have to be removed and reinserted once to the PC for the application to work.

6.3.3 Running the application

1. Start WinAmp, the gen_chipcon_remote plug-in will automatically be loaded.

2. Make sure the CC2510 USB Dongle is plugged into a USB port on the PC.

3. Turn on the SmartRF04EB development card, with the CC2510EM.

- The song title, volume, time and status will appear in the LCD screen on the SmartRF04EB.
- The controls are the following:
    - Joystick up / down – next / previous song.
    - Joystick left / right – fast forward / backwards.
    - Push joystick – toggle play / pause.
    - Push button “S1” – toggle play list shuffling.
    - Turn pot-meter “Pot” – adjust volume

7 USB development framework

A versatile, efficient and easy to use USB development framework is available for the CC2511. The intension with this framework is to make USB development as quick and easy as possible. The framework includes a firmware framework for the embedded software on the CC2511, and a Windows USB driver framework.

In addition there is general example code and libraries available showing how to use the CC2510 and CC2511 MCU, radio and other peripherals.

7.1 CC2511 USB embedded firmware library

The CC2511 USB library is a software package aimed at making development of USB devices with the CC2511 quick and simple. It supports both vendor-specified device classes and standardized device classes such as HID and USB audio, with any number of configurations, interfaces and alternate settings. The following features are provided:

- Initialization of the USB peripheral unit, and response to reset-signaling on the bus.
- Automated response to almost all standard requests, based on data from the USB descriptor set (the SYNCH_FRAME and SET_DESCRIPTOR requests must be handled by the application).
- Automated setup of endpoint control registers, depending on the selected configuration and interface alternate settings.
- All endpoint types are supported: Setup, bulk, interrupt and isochronous.
- A simple and efficient interface for responding to class and vendor requests. Both automated and manual data transfer is supported.
- Endpoint access via a simple set of macros and functions.
- Support for suspend mode and remote wake-up.
- A function hook- and event-based interface allows for the framework to run from main, the USB interrupt, or a combination of both.

The library is completely generic, flexible, efficient, and easy to use. However, it does require brief knowledge of the USB specification, including the concepts of configurations, interfaces and endpoints, different endpoint types, standard, vendor and class requests, and USB descriptor sets.

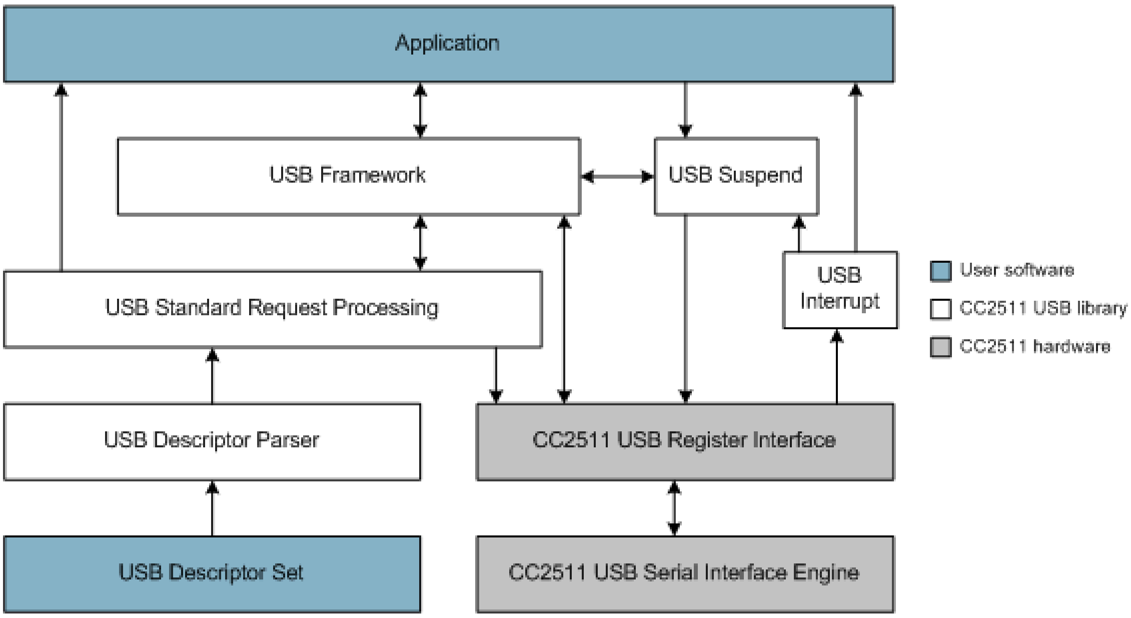


Figure 10: USB library

As shown in Figure 10 the USB library is divided into five modules:

The USB Framework (usbfw) contains all USB status information and functions for initialization, device reset handling, and transfers on endpoint 0. It also contains basic macros and functions for endpoint control and FIFO access. Vendor and class requests are passed up to the application, and standard requests further down for internal processing.

The USB Standard Requests (usbsr) module contains automated processing functions for almost all standard requests. The automated handling and endpoint configuration relies on the USB descriptor set and two look-up tables. Requests that are not supported (SYNCH_FRAME and SET_DESCRIPTOR) or that are class specific, are passed up to the application. The application is also notified upon important events, such as change in endpoint status or interface alternate settings.

The USB Descriptor Parser (usbdp) module provides a mechanism for locating standard-formatted USB descriptors. The user must provide a USB descriptor set, and two simple lookup tables - one for locating other descriptor formats, and another for setting up endpoint double-buffering. The necessary constant and structure definitions, and guidelines for writing compatible descriptor sets, are found in the USB Descriptor module.

The USB Suspend (usbsusp) module provides easy to use, fully automated support for USB suspend, USB resume and USB remote wakeup functionality.
To implement USB suspend and USB resume the user only need to add a short piece of code to the main loop.
USB remote wakeup is performed by a single function call.

The complete documentation of the CC2511 USB embedded firmware framework is available in a compressed HTML file, cc2511_USB_library.chm found in "CC2511DK\embedded_sw\Library\cc2511\CC2511UsbLibrary\doc".

7.2 Windows USB driver framework

CUDAL (Chipcon USB Dongle Access Library) provides an easy-to-use C++ interface to the CC2511 USB Dongle, making development of PC software easy. This framework is used to create the USB driver in the WinAmp remote control application example. (The other application examples use standard Windows drivers).

The following features are supported in CUDAL:

- The standard requests:
  - GET_DESCRIPTOR
  - GET_STATUS
  - SET_CONFIGURATION
  - GET_CONFIGURATION
  - SET_INTERFACE
  - GET_INTERFACE
- All vendor requests
- All class requests
- BULK and INTERRUPT transfers, with asynchronous and synchronous read and write functions

The following features are not supported in CUDAL:

- The following standard requests:
  - SET_DESCRIPTOR
  - SET_FEATURE
  - CLEAR_FEATURE
- Isochronous transfers

The complete documentation of the CUDAL framework is available in a compressed HTML file, “cudal.chm”, found in “CC2511DK\pc_sw\CUDAL\doc”

7.2.1 Windows USB driver license

The Windows USB driver framework CUDAL is based on the USBIO driver from Thesycon. This driver is not free and requires a licence.

Texas Instruments have a license that allows its customers to do the following:

- Use the application examples included in the kit.
- Develop own applications using CUDAL (Chipcon USB Dongle Access Library) as long as all development and testing is done on the hardware (CC2511 USB Dongle) supplied in the development kit from Texas Instruments.
- Modify or add functionality in the CUDAL library.

The customer must purchase their own license if they wish to do any of the following:

- Develop or test software on hardware that was not supplied in the development kit.
- Go into production with a product that uses the driver.

The customer may not:

- Change, decompile, disassemble or in any way reverse engineer the cudal.sys file.

Please contact Thesycon, www.thesycon.com, for information on how to obtain a licence.

7.3 Other firmware modules and libraries

To ease the firmware development there are also general firmware modules and libraries available for the CC2510 and CC2511.

The most important is the HAL library which contains functions and macros to enable and use all peripheral units on the CC2510 and CC2511. There is also an EB library containing functions and macros to use the peripheral units found on the SmarRF04EB development card. These libraries are all documented in the HAL_CUL_EB_Software_User_Manual.pdf.

Examples on how to use these libraries are found in the CC2510 application examples. These application examples are documented in CC2430DK_CC2510_CC1110_User_Manual.pdf. The software is found in “CC2510DK\App_Ex\cc2510”

The CC2511 application examples use a library called CC2511AppExLibrary. This library contains the following modules:

- RF module, a simple RF protocol used in the CC2511 application examples
- Timer4 manager module, use timer4 to call functions at regular intervals.
- Buffer module, implementation of a FIFO (First In, First Out) buffer.

This library is documented in a compressed HTML file, “CC2511AppExLibrary.chm”.
The CC2511AppExLibrary is divided into two parts, one for the CC2510 EM and one for the CC2511 USB Dongle.
The CC2511AppExLibrary and documentation is found here:

- CC2510 : “CC2511DK\embedded_sw\Library\cc2510\CC2511AppExLibrary”
- CC2511 : “CC2511DK\embedded_sw\Library\cc2511\CC2511AppExLibrary”

8 CC2511 Dongle hardware description

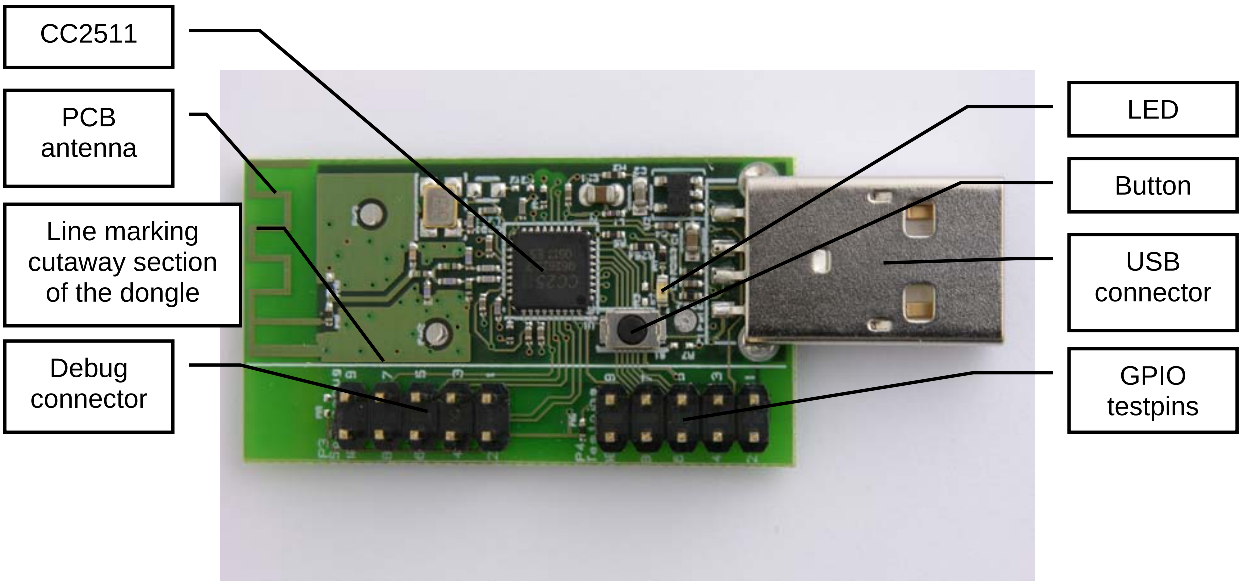


Figure 11: CC2511 Dongle top view

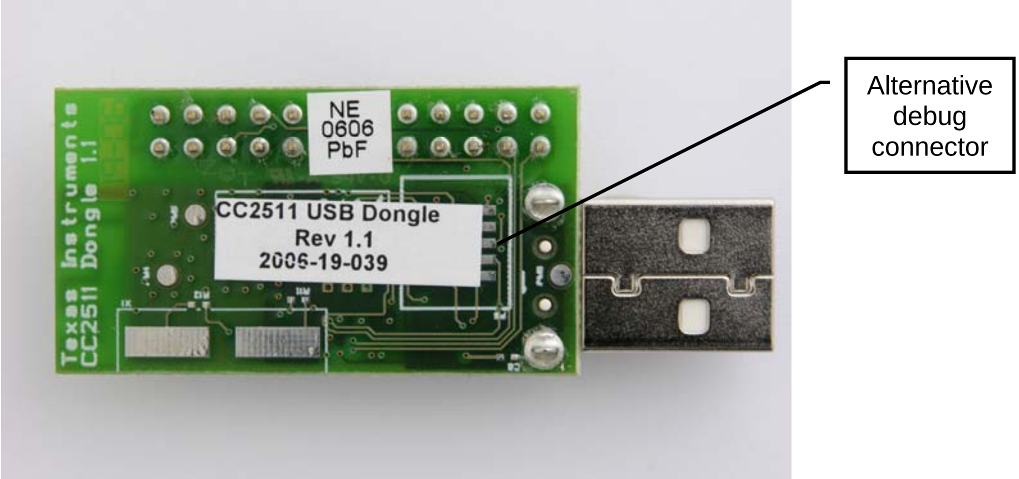

Alternative debug connector
NE 0606 PbF
CC2511 USB Dongle
Rev 1.1
2006-19-039
Texas Instruments CC2511 Dongle 1.1

Figure 12: CC2511 Dongle back view

8.1 User Interface

The CC2511 Dongle has one button and one LED that can be used to interact with the user. The LED is connected to pin P1_1. Setting P1_1 high (to logic 1) will turn the led on.
The button is connected to pin P1_2 and GND. The pin P1_2 should be configured as an input with pull-up. The button is pushed when P1_2 is pulled low (to logic 0).

8.2 USB interface

The CC2511 Dongle has a standard USB connector. The dongle is powered through this connector. On the D+ line there is a pull-up used to identify the dongle as a full-speed USB unit on the USB bus. The pull-up can be controlled from the firmware, using P1_0. This enables the firmware to be connected to the USB bus for a period of time before it signals its presence to the USB hub.

8.3 Debug connector

The CC2511 Dongle has debug connector (P3) that is compatible with the SoC debug connector on the SmartRF04EB. This enables the CC2511 Dongle to be programmed and debugged via the SmartRF04EB. The debug connector and the test pins on the CC2511 Dongle can be removed to reduce its size. If that is done, programming and debugging can still be done using the connector on the back of the CC2511 Dongle (P5). The pin-out of the debug connector (P3) is given in Table 2.

Pin	Function
1	GND
2	VDD
3	P2_2 / debug clock
4	P2_1 / debug data
5	P1_4
6	P1_5
7	RESET_N
8	P1_6
9	Not connected (VDD by mounting R6, 0 Ohm)
10	P1_7

Table 2 Debug connector (P3) pin-out

8.4 General purpose I/O connectors

All GPIO pins on the CC2511 that are not used on the Dongle are routed to the P4 “Test pins” pin-row. They can be used for debugging or to connect external devices to the CC2511. The pin-out is given in Table 3.

Pin	Function
1	P2_0
2	GND
3	P0_5
4	P0_4
5	P0_3
6	P0_2
7	P0_1
8	P0_0
9	VDD
10	P1_3

Table 3: Test pins (P4) pin-out

8.5 Cutaway section

On the CC2511 Dongle the debug connector and the GPIO test-pin connector is mounted on a section of the Dongle that may be removed to minimise the CC2511 Dongle size. This cutaway section is marked with a white line, see Figure 11. If this section is removed from the dongle, the CC2511 can still be programmed and debugged using the alternative debug connector found on the back of the CC2511 Dongle, see Figure 12.

8.6 RF Performance of PCB antenna

The performance of the PCB antenna will be affected by the device the CC2511 USB dongle is connected to. Thus, when plugged into different computers or a USB extension cable differences in the CC2511 USB dongle RF performance must be expected. The maximum antenna gain measured with this design is 5.3 dBi. Thus, duty cycling or reduction of output power might be needed to ensure compliance with regulatory limits, see application note AN032 for more information about SRD regulations in the 2.4 GHz ISM band. AN032 is available from www.ti.com/lpw or www.chipcon.com.

9 CC2511 Dongle schematics

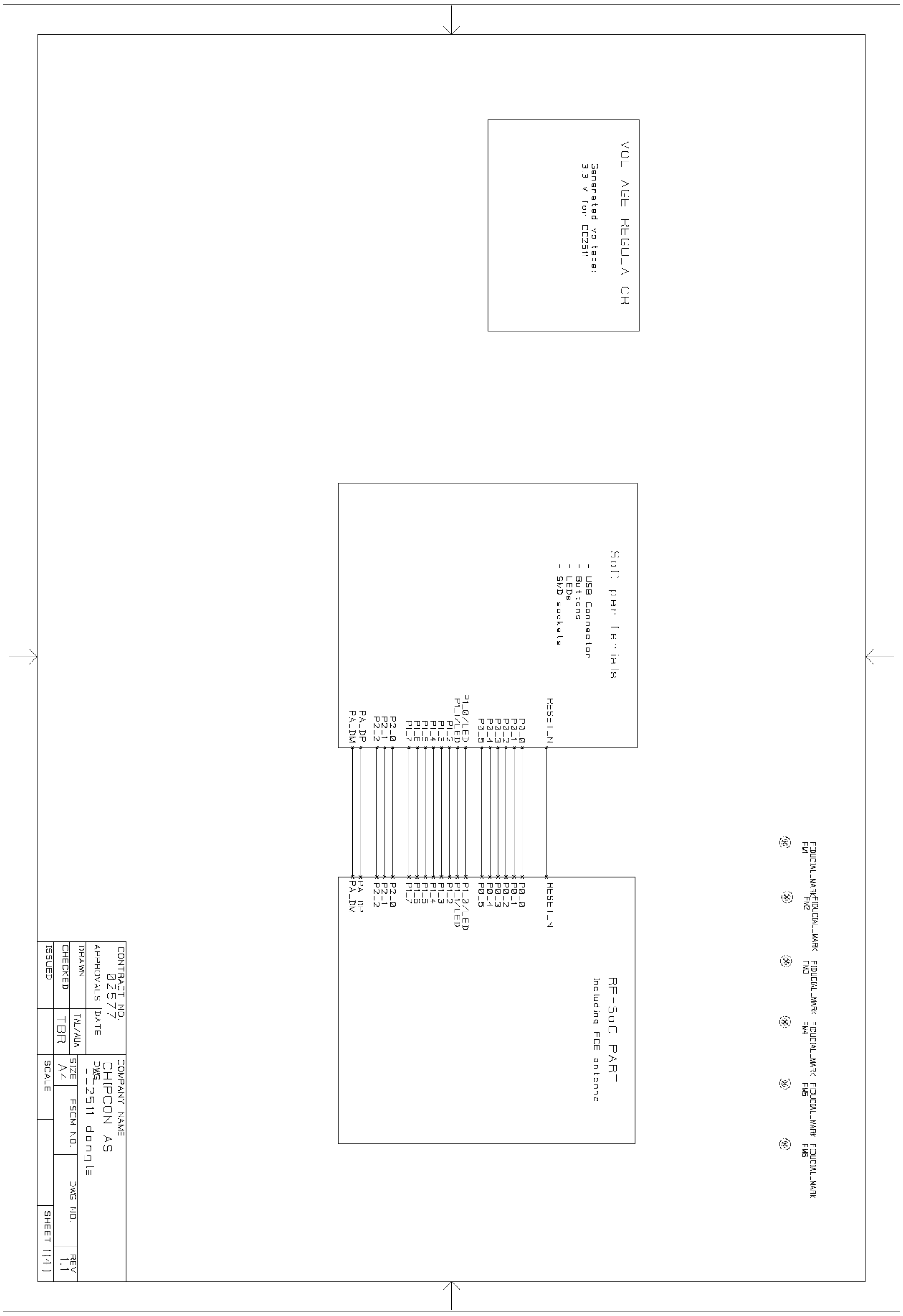


Figure 13: CC2511 Dongle schematic, page 1

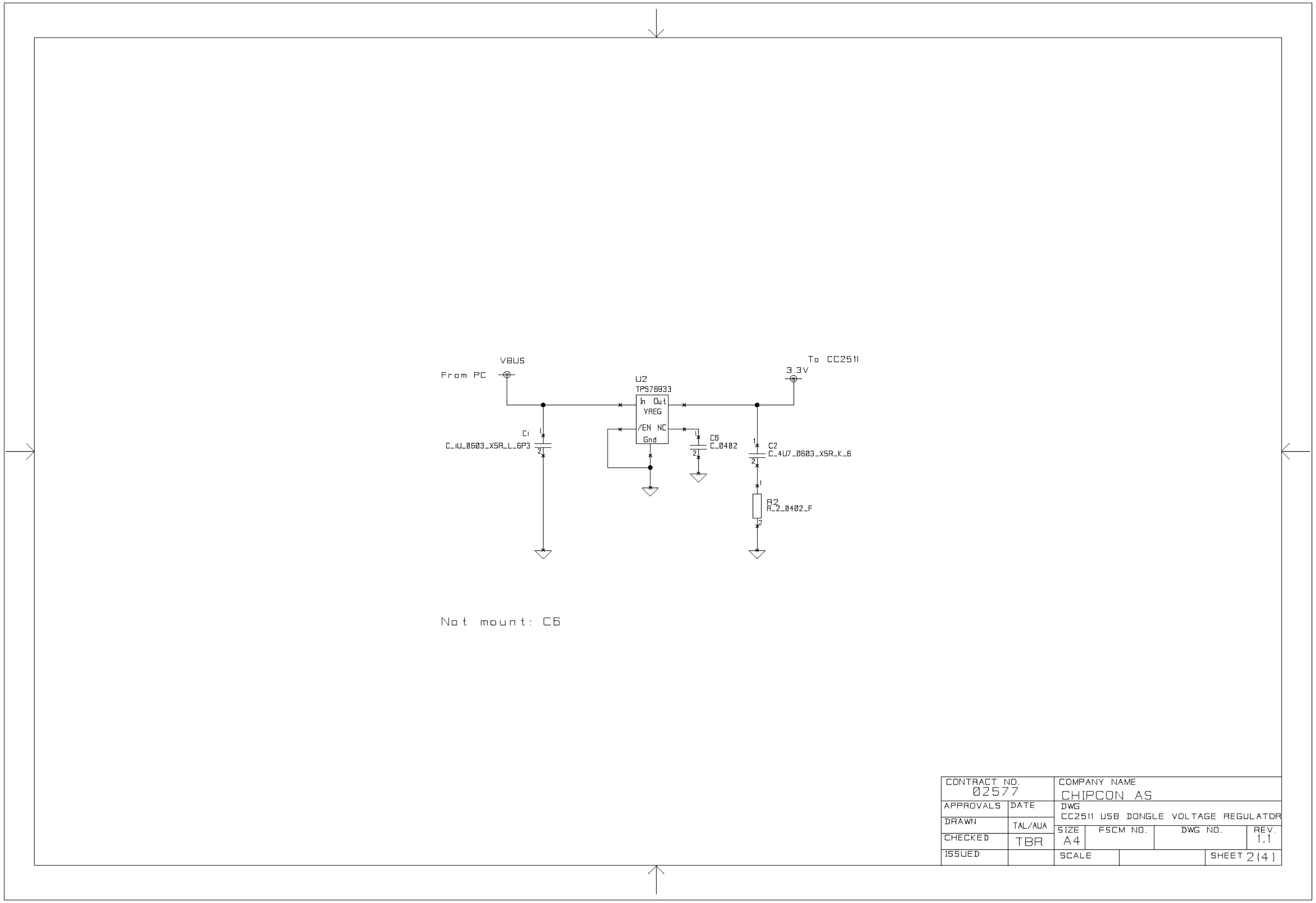


Figure 14 CC2511 Dongle schematic, page 2

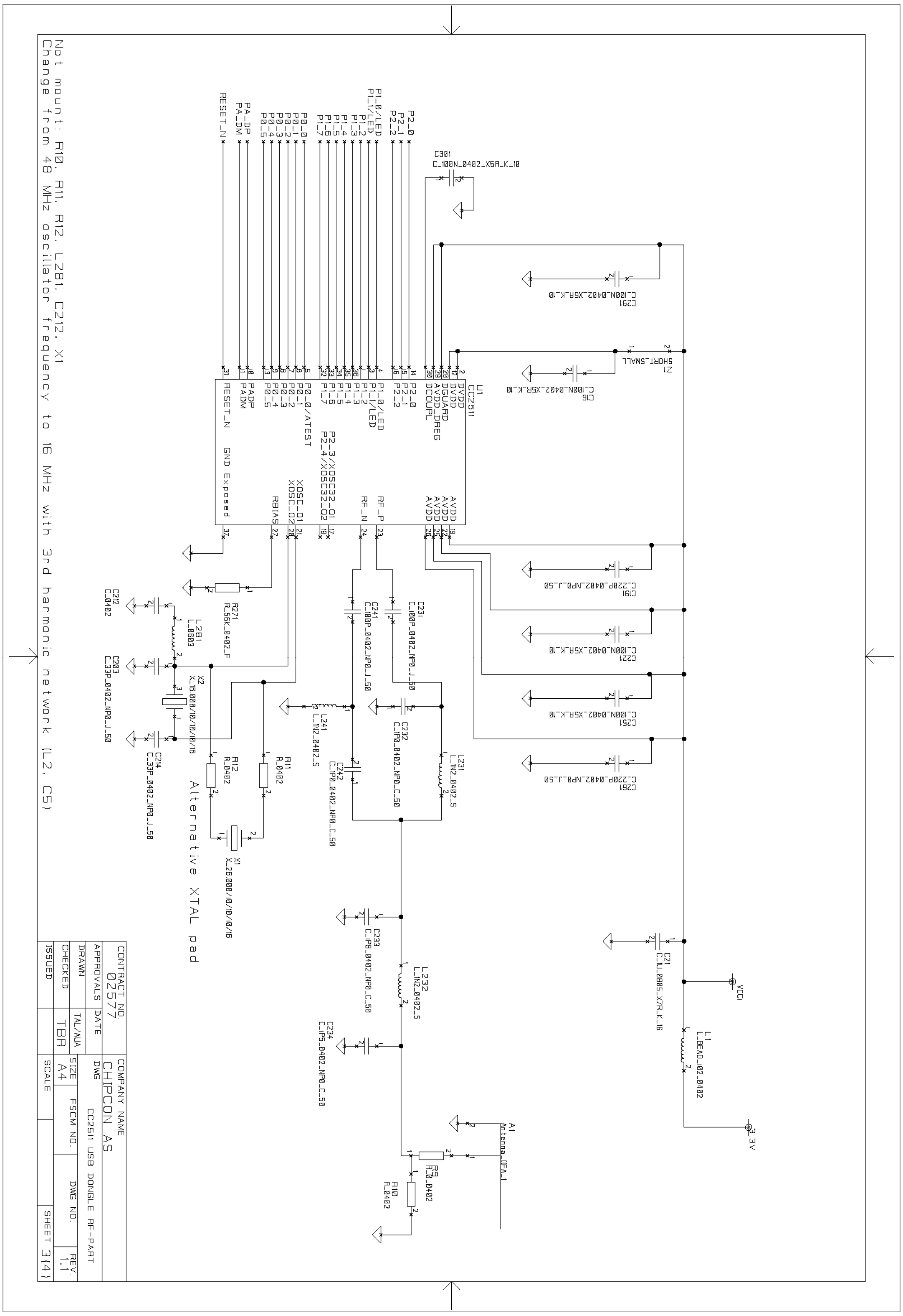


Figure 15 CC2511 Dongle schematic, page 3

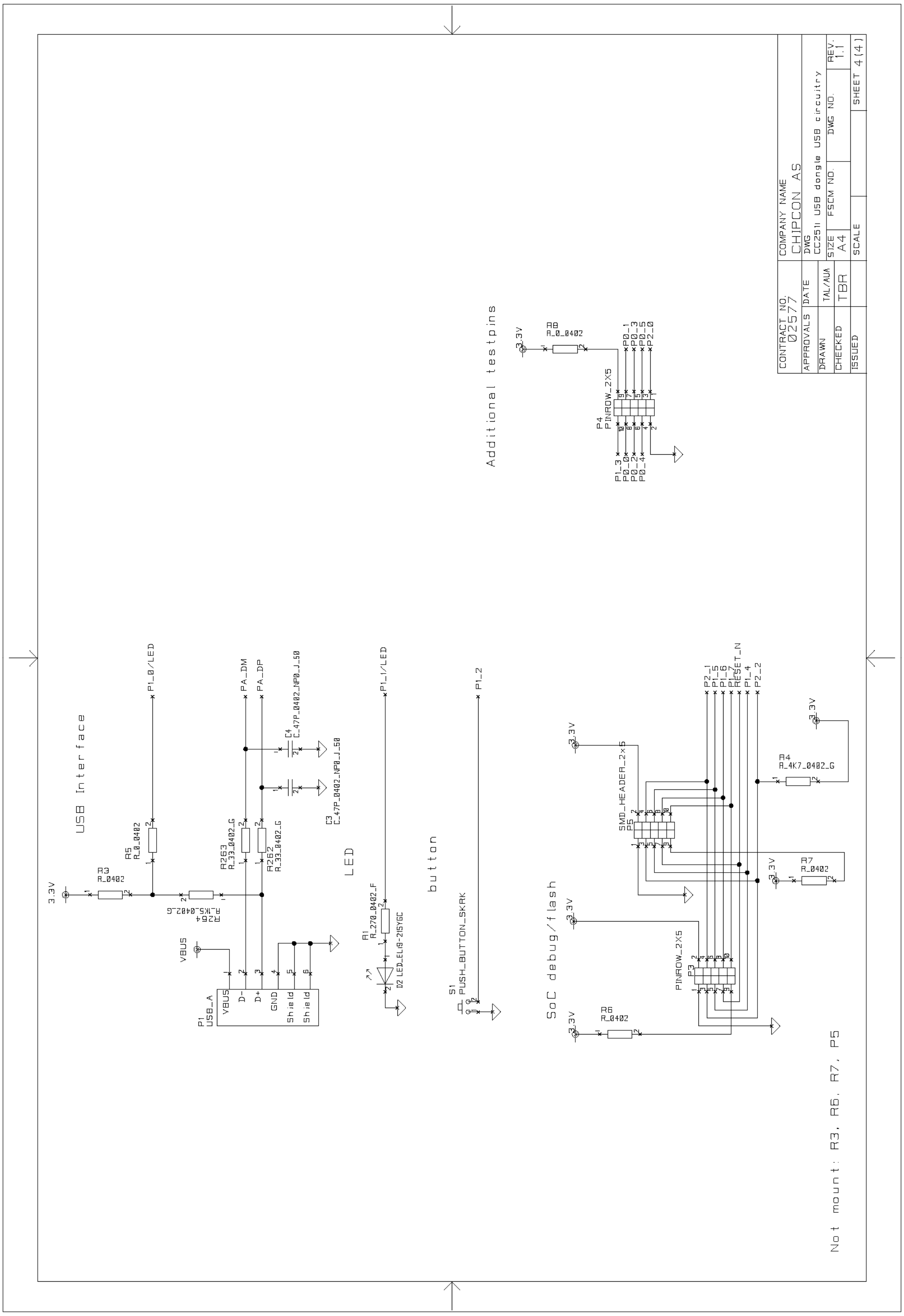


Figure 16 CC2511 Dongle schematic, page 4

10 Document history

Revision	Date	Description/Changes
1.0	2006-07-06	Initial release